REMARKS

I. General

Claims 1-29 were pending in the present application, and all of the pending claims are rejected in the current Office Action (mailed September 22, 2004). The outstanding issues raised in the current Office Action are:

- Claims 3, 5, and 17 are rejected under 35 U.S.C. § 112, second paragraph; and

- Claims 1-29 are rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,775,692 issued to Albert et al (hereinafter "*Albert*").

In response, Applicant respectfully traverses the outstanding claim rejections, and requests reconsideration and withdrawal thereof in light of the amendments and remarks presented herein.

II. Amendments

Claims 3, 5, 17, and 22 are amended herein. No new matter is added by these claim amendments.

Claim 3 is amended to provide antecedent basis for the recited second BTCP module. This is not intended to be a narrowing amendment, but is instead merely cosmetic to ensure proper antecedent basis for the recited second BTCP module.

Claim 5 is likewise amended to provide antecedent basis for the recited second BTCP module. This is not intended to be a narrowing amendment, but is instead merely cosmetic to ensure proper antecedent basis for the recited second BTCP module.

Claim 17 is amended to recite the "first" TCP module to ensure clarity regarding the TCP module to which the claim refers. Further, the element "at said front end server" is deleted. This is not intended to be a narrowing amendment.

Claim 22 is amended to replace the recitations of "front end server" with "front-end node". This is not intended to be a narrowing amendment, but is instead, if anything, a broadening amendment in that the term "node" encompasses a server. Further, claim 22 is

amended to replace "front end TCP module" with "front-end TCP module". This amendment is not intended to be narrowing, but is instead merely cosmetic to ensure consistency in using "front-end" within this claim.

III. Rejections Under 35 U.S.C. § 112, Second Paragraph

Claims 3, 5, and 17 are rejected under 35 U.S.C. § 112, second paragraph. Claim 3 is rejected because there is insufficient antecedent basis for the "said second BTCP module" recited therein. As described above, claim 3 is amended herein in a manner that corrects this deficiency.

Claim 5 is rejected because there is insufficient antecedent basis for the "said second BTCP module" recited therein. As described above, claim 5 is amended herein in a manner that corrects this deficiency.

Claim 17 is rejected because there is insufficient antecedent basis for the "said front end server" and the "the TCP module" limitations recited therein. As described above, claim 17 is amended herein in a manner that corrects these deficiencies.

In view of the above, Applicant respectfully requests withdrawal of the rejections of claims 3, 5, and 17 under 35 U.S.C. § 112, second paragraph.

IV. Rejections Under 35 U.S.C. § 102(e)

Claims 1-29 are rejected under 35 U.S.C. § 102(e) as being anticipated by *Albert*. Applicant respectfully traverses this rejection as provided further below.

To anticipate a claim under 35 U.S.C. § 102, a single reference must teach every element of the claim, *see* M.P.E.P. § 2131. Applicant respectfully submits that *Albert* fails to teach each and every element of claims 1-29.

Independent Claim 1

Albert fails to teach all elements of independent claim 1. As described further below, *Albert* does not provide a modularized solution, and thus fails to teach at least the modules recited in claim 1.

First, independent claim 1 recites "establishing a communication session between a client and a front-end node at a first bottom TCP (BTCP) module located below a first TCP module in a first operating system at said front-end node, said front-end node accessing a plurality of back-end web servers forming a web server cluster that contains content" (emphasis added). *Albert* fails to teach a first bottom TCP (BTCP) module located below a first TCP module in a first operating system at a front-end node.

The current Office Action cites col. 7, lines 36-60 of *Albert* in support of its assertion that *Albert* teaches this element of claim 1, *see* page 3 of the Office Action. Col. 7, lines 36-60 of *Albert* provides:

> FIG. 2A is a block diagram of a network architecture that provides network services without requiring a network service appliance to be physically placed at a node through which all incoming and outgoing packets processed by a group of servers must pass. Several clients 201, 202, and 203 are connected to a network 210. Network 210 is connected to a group of servers 220 that includes servers 221, 222, and 223. There is no point through which all traffic between devices connected to network 210 and the group of servers 220 must pass. Instead, some traffic from network 210 that is bound for the group of servers passes through a forwarding agent 231 and some traffic between network 210 and group of servers 220 passes though a forwarding agent 232.
>
> In the example shown, forwarding agent 231 is connected to server 221 and server 222 and forwarding agent 232 is connected to server 222 and server 223. Thus, server 222 may communicate with network 210 through either of the forwarding agents, server 221 communicates with network 210 exclusively through forwarding agent 231, and server 223 communicates with network 210 exclusively through forwarding agent 232. This arrangement may be generalized to include an arbitrary number of servers connected to an arbitrary number of forwarding agents with individual servers connected to arbitrary subsets of the forwarding agents.

The above portion of *Albert* teaches forwarding agents 231 and 232 through which traffic flows between the clients and the servers. The Office Action appears to contend that the forwarding agents of *Albert* are front-end nodes, as recited by claim 1. However, *Albert* in no way teaches that either of the forwarding agents 231 and 232 include a first bottom TCP

(BTCP) module located below a first TCP module in a first operating system. *Albert* does not teach a module at all in the above relied upon portion, and certainly not a BTCP module located below a TCP module in an operating system.

Additionally, independent claim 1 further recites "b) receiving a HTTP request from said client at said first BTCP module; c) parsing said HTTP request to determine which back-end web server, a selected back-end web server, in said plurality of back-end web servers can process said HTTP request, said selected back-end web server not said front-end node" (emphasis added). *Albert* fails to teach parsing a received HTTP request to determine which of a plurality of back-end web servers can process the HTTP request.

The Office Action cites col. 9, lines 10-34 and 45-58 of *Albert* in support of its assertion that *Albert* teaches this element of claim 1, *see* page 3 of the Office Action. Col. 9, lines 10-34 and 45-58 of *Albert* provides:

> In addition to specifying instructions for each flow, service managers must also obtain information about each new flow from the forwarding agents. For example, when a service manager provides load balancing through a set of forwarding agents, the service manager uses fixed affinities to provide specific instructions to the forwarding agents detailing where packets for each load balanced flow are to be forwarded. In addition to providing those specific instructions, the service manager also provides general instructions to each forwarding agent that specify which new flows the service manager is interested in seeing. These general instructions are provided using wildcard affinities. Wildcard affinities, which are described in detail below, specify sets of flows that are of interest to a service manager. In one embodiment, this is done by specifying subnet masks that determine sets of source and destination IP addresses that will be forwarded to a service manager. In addition, ports or sets of ports and protocol may be specified in wildcard affinity as well. As is described further below, the use of wildcard affinities enables separate service managers to be configured to provide services for different sets of flows. Each service manager specifies the flows of interest to it and other service managers handle other flows. In this manner, service managers can be configured in parallel to share load.
>
> * * *
>
> In the case of load balancing, service managers send wildcard affinities to forwarding agents. The wildcard affinities specify destination IP addresses that correspond to virtual IP addresses of server clusters that are to be load balanced by the service manager. The forwarding agents then forward new packets sent to those virtual IP addresses to the appropriate service manager. The service manager selects a server from the server cluster and then the service manager sends a fixed affinity to each forwarding agent that instructs

> the forwarding agent to forward packets for that specific flow to the selected server in the cluster. Forwarding agents may also forward packets for purposes other than load balancing. Packets may be forwarded to real IP addresses as well as virtual IP addresses.

Albert does not teach parsing a received HTTP request to determine which back-end web server can process such HTTP request. Rather, *Albert* teaches pre-setting a wildcard affinity based on a client's IP address. For instance, the above portion of *Albert* teaches that wildcard affinities specify sets of flows that are of interest to a service manager. The wildcard affinities are pre-selected (e.g., before even receiving a request from a client), to specify a subnet mask that identifies a set of source and destination IP addresses. Thus, for instance, a particular IP address corresponding to a client of interest can be identified by a wildcard affinity.

As described further in *Albert* at col. 12, line 6 – col. 14, line 15, a Syn packet is used to identify whether the flow matches a wildcard affinity, in which case it is forwarded to the service manager for determination of how to handle the flow. Thus, the service manager in *Albert* selects a back-end server responsive to receipt of a Syn packet, which is an initial connection establishment packet sent before it is even known what the request will be. That is, the Syn packet upon which *Albert* selects a server, does not include an HTTP request. Rather, the Syn packet includes source and destination IP addressed, from which it is determined by the forwarding agents whether such packet corresponds to a pre-set wildcard affinity. Thus, the back-end server is not selected in *Albert* as a result of parsing a received HTTP request, as the back-end server is selected based on other information (e.g., source IP address) included in the Syn packet.

Further, claim 1 recites "switching a bottom IP (BIP) module at said front-end node to a forwarding mode, wherein packets received at said BIP module from said client are forwarded to said selected back-end web server, said BIP module located below an IP module at said front-end node" (emphasis added). As described above, *Albert* does not teach modules. Specifically, *Albert* does not teach a BIP module located below an IP module at the front-end node (e.g., at the forwarding agent of *Albert*).

The Office Action cites col. 14, lines 1-15 of *Albert* in support of its assertion that *Albert* teaches this element of claim 1, *see* page 4 of the Office Action. Col. 14, lines 1-15 of *Albert* provides:

> Client 304 sends a data packet to forwarding agent 302. Forwarding agent 302 has stored the fixed affinity corresponding to the flow from the client to the host in a fixed affinity database 303. Forwarding agent 302 notes the match of the 5-tuple of the data packet with an affinity key in the fixed affinity database and then forwards the data packet according to the action defined in that fixed affinity. In this example, the action defined is to translate the destination IP address of the client from the virtual IP address of virtual machine 310 to the IP address of host 306. In addition to forwarding the data packet, the affinity found by the forwarding agent also includes an action that requires the forwarding agent to send an affinity packet to service manager 300 that includes data about the packet for the purpose of service manager 300 gathering statistics about network traffic.

The above portion of *Albert* does not teach a module, and particularly not a BIP module located below an IP module at said front-end node, as recited by claim 1. Instead, the above portion of *Albert* simply teaches that the client sends a data packet to the forwarding agent, and the forwarding agent translates (using the affinity key) the destination IP address of the client from the virtual IP address of the virtual machine to the IP address of a specific host. The forwarding agent also sends an affinity packet to the service manager so that the service manager can gather statistics about network traffic. There is no teaching or even a hint of a BIP module located below an IP module at the front-end node (e.g., the forwarding agent). The forwarding actions performed by *Albert*, including the forwarding agent accessing the affinity database to determine a forwarding action based on a match in the database with the affinity key, are simply not taught as being performed via modules. Thus, the solution of *Albert* suffers at least from the drawbacks identified in the present application as associated with non-modularized implementations.

Accordingly, *Albert* fails to teach at least the above-identified elements of claim 1, and therefore claim 1 is not anticipated under 35 U.S.C. § 102 by *Albert*.

Independent Claim 11

Albert fails to teach all elements of independent claim 11. As described further below, *Albert* does not provide a modularized solution, and thus fails to teach at least the modules recited in claim 11.

First, independent claim 11 recites "establishing said communication session between said client and said first BTCP module, said first BTCP module located below a first TCP module in a first operating system at said front-end node" (emphasis added). As described above with claim 1, *Albert* fails to teach a first bottom TCP (BTCP) module located below a first TCP module in a first operating system at a front-end node.

Additionally, independent claim 11 further recites "c) receiving a HTTP request from said client at said first BTCP module; d) parsing said HTTP request to determine which back-end web server, a selected back-end web server, in said plurality of back-end web servers contains said data in order to process said HTTP request, said selected back-end web server not said front-end node" (emphasis added). As described above with claim 1, *Albert* fails to teach parsing a received HTTP request to determine which of a plurality of back-end web servers can process the HTTP request.

Further, claim 11 recites "switching a bottom IP (BIP) module in said front-end node to a forwarding mode, wherein packets, from said client, received at said front-end node are intercepted by said BIP module and forwarded to said selected back-end web server, said BIP module located below an IP module in said front-end node, said BIP module changing destination IP addresses of said packets to said selected back-end web server" (emphasis added). As described above with claim 1, *Albert* does not teach modules and specifically does not teach a BIP module located below an IP module at the front-end node (e.g., at the forwarding agent of *Albert*).

Accordingly, *Albert* fails to teach at least the above-identified elements of claim 11, and therefore claim 11 is not anticipated under 35 U.S.C. § 102 by *Albert*.

Independent Claim 22

Albert fails to teach all elements of independent claim 22. As described further below, *Albert* does not provide a modularized solution, and thus fails to teach at least the modules recited in claim 22.

First, independent claim 22 recites "a front-end node coupled to said client by said communication network, said front-end node including a front-end bottom TCP (BTCP) module located below a front-end TCP module in a first operating system, and a bottom IP (BIP) module located below an IP module in said first operating system" (emphasis added). As described above with claim 1, *Albert* fails to teach a front-end node that includes modules. Particularly, *Albert* does not teach that its forwarding agents include a front-end bottom TCP (BTCP) module located below a front-end TCP module in a first operating system. Further, *Albert* does not teach that its forwarding agents include a bottom IP (BIP) module located below an IP module in the first operating system. Thus, *Albert* fails to teach at least this element of claim 22.

Further, claim 22 recites "a plurality of back-end web servers including a selected back-end web server, said plurality of back-end web servers containing content that is partitioned between each of said plurality of back-end web servers, each of said plurality of back-end web servers coupled to said front-end node through said communication network, each of said plurality of back-end web servers including a back-end bottom TCP module located below a back-end TCP module" (emphasis added). *Albert* does not teach back-end web servers that include modules, and particularly not back-end web servers that include a back-end bottom TCP module located below a back-end TCP module. Thus, *Albert* fails to teach at least this element of claim 22.

Accordingly, *Albert* fails to teach at least the above-identified elements of claim 22, and therefore claim 22 is not anticipated under 35 U.S.C. § 102 by *Albert*.

Dependent Claims

In view of the above, Applicant respectfully submits that independent claims 1, 11, and 22 are not anticipated under 35 U.S.C. § 102 over *Albert*. Further, each of dependent

claims 2-10, 12-21, and 23-29 depend either directly or indirectly from one of independent claims 1, 11, and 22, and thus inherit all limitations of the respective independent claim from which they depend. It is respectfully submitted that dependent claims 2-10, 12-21, and 23-29 are allowable not only because of their dependency from their respective independent claims for the reasons discussed above, but also in view of their novel claim features (which both narrow the scope of the particular claims and compel a broader interpretation of the respective base claim from which they depend).

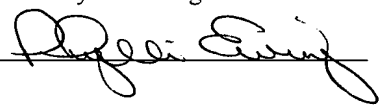
V. Conclusion

In view of the above, Applicant believes the pending application is in condition for allowance. Applicant believes no fee is due with this response. However, if a fee is due, please charge our Deposit Account No. 08-2025, under Order No. 10012351-1 from which the undersigned is authorized to draw.

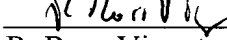
I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail, Label No. EV 482745211US in an envelope addressed to: M/S Amendment, Commissioner for Patents, Alexandria, VA 22313.

Date of Deposit: December 22, 2004

Typed Name: Phyllis Ewing

Signature: 

Respectfully submitted,

By: 

R. Ross Viguet
Attorney/Agent for Applicant(s)
Reg. No. 42,203
Date: December 22, 2004
Telephone No. (214) 855-8000